59JTagStick 取扱説明書

(59JTagStick-8)

Rev1.05 2015.04.17(ROMVerJ1.906)

Rev1.04 2015.04.12(ROMVerJ1.905)

Rev1.03 2015.03.02(ROMVer.J1.904b)

Rev1.02 2015.01.22(ROMVer.J1.904)

Rev1.01 2014.12.12(ROMVer.J1.902)

Rev1.00 2014.12.02(ROMVer.J1.902)

悟空株式会社

改訂履歴

日時	担当者	Rev	改訂内容概要
2014/12/01	大庭	1.00	初版
2014/12/12	大庭	1.01	誤記修正、6.5.5 DaisyChain と in-line コマンドについて　追加。
2015/1/22	大庭	1.02	
2015/3/2	大庭	1.03	JTAG で使わないコマンドの削除 6.5.SVF ファイルの対応、6.6.reformSVF の使い方　を追加 6.10. “CONFIG.TXT”サンプルの場所移動、説明追加。
2015/4/12	大庭	1.04	図 8.3 不明瞭のため差替 4.6. XODE ピン　機能割り当て 5.1. LED 点灯条件(残量表示)、SVF_interpreter エラー表示追加 5.2.　転送残量表示機能　追加 5.3. SVF エラー表示機能　追加 5.4.　バージョン表示機能　一時的に機能削除 6.3. Commands (c)#S:　周波数割り当て変更 6.3. Commands (n)⑩#P : C9　一時削除 6.3. Commands (n)⑪#P : CA　追加 6.3. Commands (n)⑫#P : CE　追加 6.6. reformSVF.exe の使い方　説明書を分離
2015/4/17	大庭	1.05	1.1. 59JTagStick の機能・特徴　TCK の分周の記述修正

目次

1. 機能･特徴 5
1.1. 59JTagStick の機能・特徴 5
2. 形状 7
3. 電気特性 8
3.1. 絶対定格 8
3.2. 推奨動作範囲 8
3.3. 消費電流 8
4. Pin 機能表 9
4.1. 本体ピン機能表 9
4.2. ALTERA 用変換コネクタ ピン機能表 10
4.3. XILINX 用変換コネクタピン機能表 11
4.4. uSDCONF1A03 ソケットピン機能表 12
4.5. XAREA 詳細 13
4.6. XMODE ピン 14
5. LED 14
5.1. LED の点灯条件 15
5.2. 転送残量表示機能 16
5.3. SVF エラー表示機能 17
5.4. バージョン情報表示機能 18
6. “CONFIG.TXT”の準備 19
6.1. “CONFIG.TXT”とは 19
6.2. Commands 速見表 20
6.3. Commands 21
6.4. DaisyChain と in-line コマンドについて 28
6.4.1. DaisyChain の表記方法 28
6.4.2. In-line コマンドの表記方法 29
6.4.3. FPGA をバイパスする方法 30
6.4.4. In-Direct モードの指定方法 31
6.5. SVF ファイルの対応 32
6.5.1. SVF ファイルの対応 32
6.5.2. SVF ファイルの対応コマンド 32
6.5.3. GVF ファイルへの変換の必要性 33
6.5.4. SVF,GVF ファイルの記述例 34
6.6. reformSVF.exe の使い方 35

6.7. D1,D2,PO の設定について 36
6.7.1. D1 パラメータ 36
6.8. コマンドパラメータ　デフォルト値......38
6.9. RBF データの生成方法......40
6.10. “CONFIG.TXT”サンプル 41
6.10.1. アルテラの場合 41
6.10.2. XILINX の場合 42
7. DCard の活線挿抜について 43
7.1. 構成図 43
7.2. SDCard の活線挿抜対策......44
8. モジュールのソケットからの取り外しについて 45
8.1. モジュール裏面 45
8.2. モジュールの取り外し方......45
9. 添付品 47
10. ゲートサイズ、VCCIO 電圧とシールの色 48
11. 制約事項 49
12. アップデート機能(順次対応予定) 50

1. 機能･特徴

1.1. 59JTagStick の機能・特徴

- 本モジュールは uSDCONF1A03JTag モジュールを、JTAG ポートの VCCIO 電源から、モジュール本体を駆動させるのに必要な電源を生成し、JTAG ポートの各信号レベルを VCCIO に変換する機能を有したマザーボードとの組みあわせによる複合モジュールです。
- VCCIO=1.2V から 3.6V まで動作します。
- microSD カードには、最大 2GB までの FAT16 注1)でフォーマットされた Card を使用でき、容量の許す限り注2)バイナリデータを格納することができます。
- microSD カードへのバイナリデータの書き込みは、Windows システムの Explorer 上で Drag & Copy するだけで簡単に行えます。 JTag ケーブルを介してパソコンと接続する煩わしさがありません。
- microSD カード上に格納する“CONFIG.TXT”の名称のアスキーキャラクタで記述された制御ファイルにより、FPGA に転送するバイナリデータの選択や、コンフィギュレーションに関する信号間の遅延量など、各種パラメータを指定することができ、専用のソフトを使うことなく、使い慣れたテキストエディタで編集するだけです。
- “CONFIG.TXT”の先頭で、メーカ指定コマンドを入力すると、そのメーカに固有のパラメータ値をデフォルトとして設定することが可能です。
- microSD カード上に格納されたバイナリデータは、“CONFIG.TXT”上で、バイナリデータ名と AREA[3:0]ピンによって指し示される 0 から F の 16 個の 16 進数文字と関連付けを行うことにより、AREA[3:0]ピンに接続されたロータリーSW 設定により、最大 16 個のバイナリデータを瞬時に選択し、FPGA をコンフィギュレーションすることが可能です。(例えば #3 : data3.rbf で SW 設定3と data3.rbf を関連付けします。)
- 本機能は、例えばデバッグの場面で開発段階の複数のバージョンのバイナリデータの動作比較や、デモの場面で複数のアプリケーションを瞬時に切り替えてデモンストレーションする場面などで威力を発揮します。
- microSD カードとのインタフェースは 50MHz の HighSpeed モードで転送を行います。
- FPGA への転送クロック TCK は、“CONFIG.TXT”で指定することにより、下記の速度から選択可能です。
  50MHz(デフォルト),25MHz, 12.5MHz, 6.25MHz, 3.57MHz,2.778MHz, 1.923MHz,1MHz
  (現在、50MHz 系列のみのサポートです。60MHz 系列は後日サポートの予定です。)
- バイナリデータの先頭にプリアンブルを挿入することや、バイナリデータの最後にポストアンブルを可変長で挿入することが可能です。
- リセット解除から TCK 送信開始までの間隔を D2 パラメータにより設定することが可能です。

  注 1)ファイル名は最大8文字＋3文字までで、EFAT には対応していません。
  注 2)ルートディレクトリに格納するため、FAT の構成上、最大 512 個の上限があります。

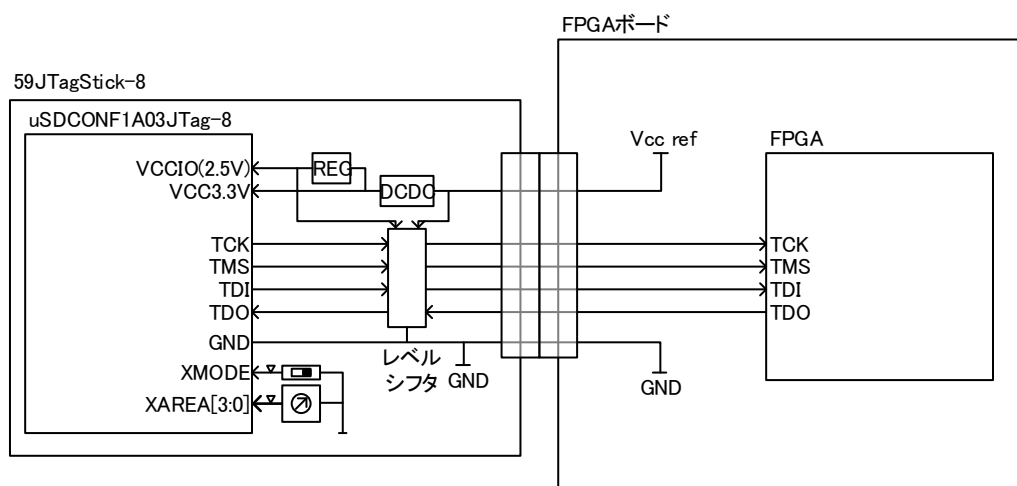



図 1.1 59JTagStick ブロックイメージ図

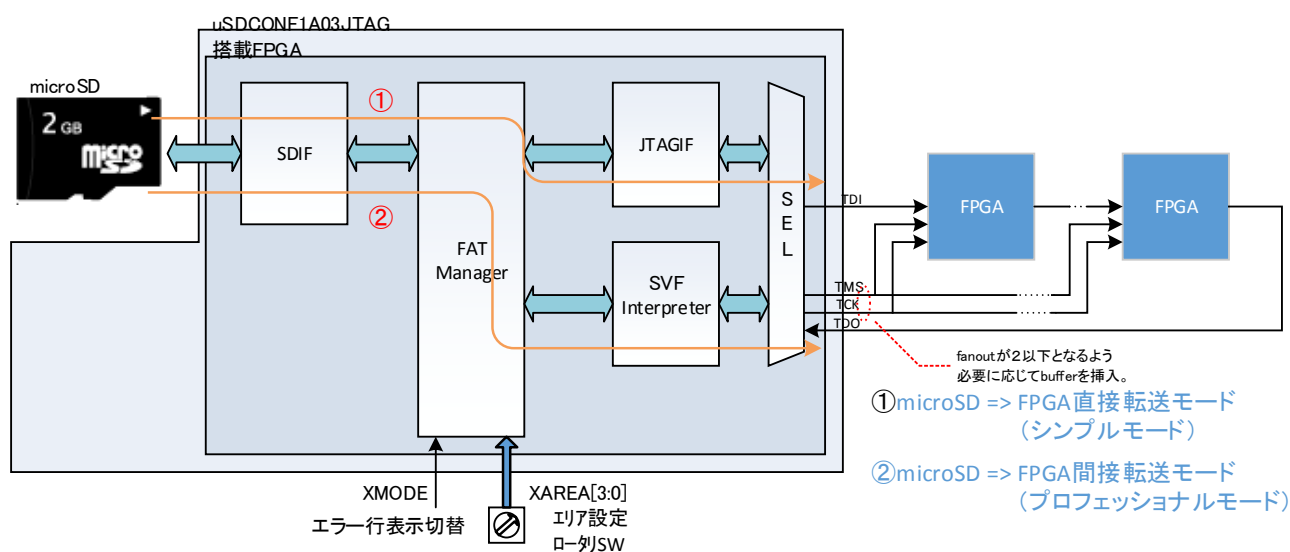



図 1.2 uSDCONF1A03JTAG ブロックイメージ図

2. 形状

59JtagStick とそれに実装される uSDCONF1A03JTAG モジュールの形状を掲載します。

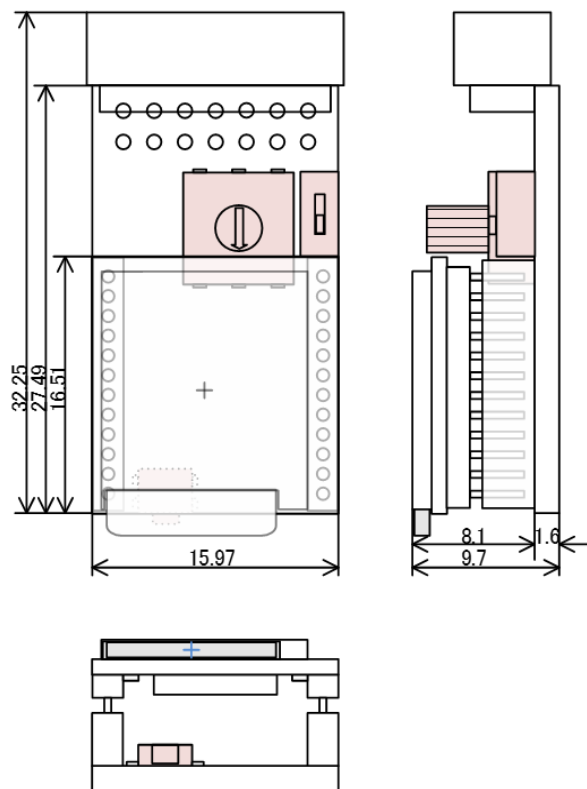



図 2.1 59JTagStick 形状図

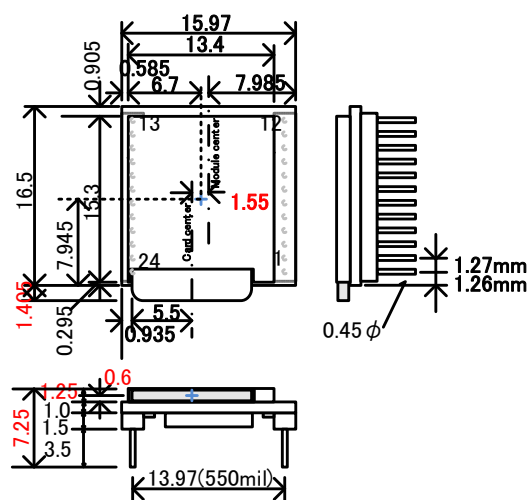



図 2.2 uSDCONF1A03JTAG 形状図

3. 電気特性

3.1. 絶対定格

Supply Voltage VCC3V	-0.2V	to	3.75V
Input or I/O Tristage Voltage Applied	-0.2V	to	3.75V
Storage Temperature(Ambient)	-65 ℃	to	150 ℃

3.2. 推奨動作範囲

Supply Voltage VCC3V	1.2V	to	3.6V
Ambient Temperature	0 ℃	to	70 ℃

3.3. 消費電流

品目		Typcal(実測)	Max	備考
Module 本体	VCC(3.3V)		90mA	
	VCCIO(3.3V 時)		10mA	
microSD カード	VDD(3.3V)		100mA	規格上の Max 値
合計			200mA	

表 3.3 消費電流表

※消費電流は ROM のバージョンによって変化する場合があります。

4. Pin 機能表

4.1. 本体ピン機能表

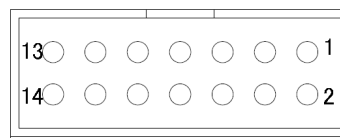
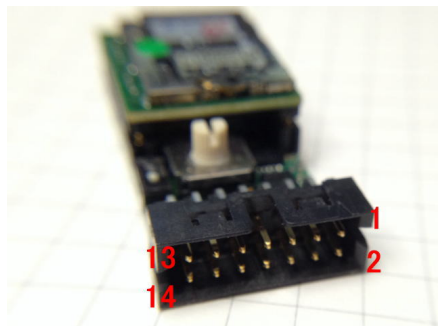
59JTagStick モジュールのピン機能表を掲載します。

Pin #	信号名	Dir	備考
2	VCCIO	PU	
4	TMS	O	
6	TCK	O	
8	TDO	I	
10	TDI	O	
12	nSTATUS(INITB)	I	未使用
1,3,5,7,9,11,13	GND	PU	

表 4.1 59JTagStick ピン機能表

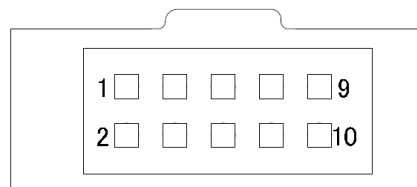
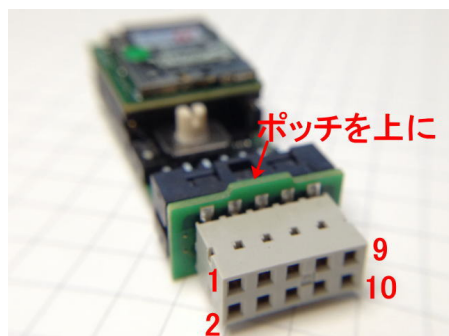
4.2. ALTERA 用変換コネクタ ピン機能表

Pin #	信号名	Dir	備考
1	TCK	O	
2	GND		
3	TDO	I	
4	VCCIO		
5	TMS	O	
6	-		
7	nSTATUS(INITB)	I	未使用
8	-		
9	TDI	O	
10	GND		

表 4.2 ALTERA 変換コネクタ ピン機能表

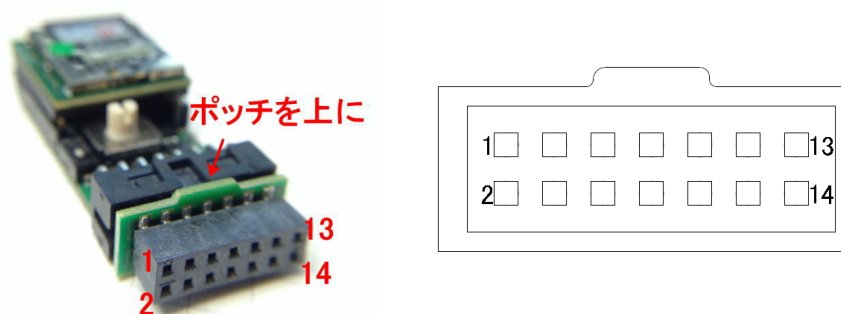
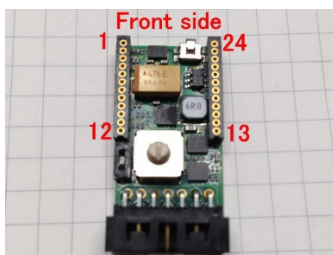
4.3. XILINX 用変換コネクタピン機能表

Pin #	信号名	Dir	備考
2	VCCIO	PU	
4	TMS	O	
6	TCK	O	
8	TDO	I	
10	TDI	O	
12	nSTATUS(INITB)	I	未使用
1,3,5,7,9,11,13	GND	PU	

表 4.3 XILINX 変換コネクタ ピン機能表

4.4. uSDCONF1A03 ソケットピン機能表

uSDCONF1A03JTag モジュール実装用のソケットのピン配置は下図の通りです。

水平 1.27mm ハーフピッチ実装タイプのピン機能表を示します。

Pin No.	Signal name	Dir	内部 Pullup	Descriptions
1	未使用	O		未使用(High-Z 出力＋内部 Pullup)　(XCS0)
2	VCCIO	ー		入出力信号の VCCIO 電源入力。
3	XAREA0	I	4.7K	バイナリデータ選択用ロータリーSW(LSB)
4	VCC3V	ー		3.3V 電源入力
5	XAREA1	I	4.7K	バイナリデータ選択用ロータリーSW
6	GND	ー		GND
7	XAREA2	I	4.7K	バイナリデータ選択用ロータリーSW
8	TDO	I	330Ω	JTAG TDO 信号入力　(DONE)
9	TCK	O		JTAG TCK 信号出力　(CCLK)
10	TMS	O	4.7K	JTAG TMS 信号出力　(PROGB)
11	XAREA3	I	4.7K	バイナリデータ選択用ロータリーSW(MSB)
12	INITB(未使用)	O	4.7K	未使用(High-Z 出力＋4.7K Pullup)　(INITB)
13	XRST	I	4.7K	パワーオンリセット信号
14	TDI	O		JTAG TDI 信号出力　(D0)
15	未使用	O		未使用(High-Z 出力＋内部 Pullup)　(D1)
16	未使用	O		未使用(High-Z 出力＋内部 Pullup)　(D2)
17	未使用	O		未使用(High-Z 出力＋内部 Pullup)　(D3)
18	未使用	O		未使用(High-Z 出力＋内部 Pullup)　(D4)
19	GND	ー		GND
20	未使用	O		未使用(High-Z 出力＋内部 Pullup)　(D5)
21	未使用	O		未使用(High-Z 出力＋内部 Pullup)　(D6)
22	未使用	O		未使用(High-Z 出力＋内部 Pullup)　(D7)
23	XMODE	I	4.7K	GVF ファイルのエラー検出行の表示の有無指定
24	未使用	O		未使用(High-Z 出力＋内部 Pullup)　(XCS1)

表 4.4 uSDCONF1A03JTag ピン機能表

注1)　XAREA[3:0],XMODE, DONE,nCONFIG,nSTATUS の信号には内部に Pullup 抵抗が実装されており、外部で実装する必要はありません。

4.5.　XAREA 詳細

- “CONFIG.TXT”ファイル上で microSD カード上に格納されたバイナリデータとマザーボード上のロータリースイッチに接続された AREA[3:0]ピンによって指し示される 0～F の 16 個の 16 進数文字と関連付けがなされます。
- XAREA[3:0]入力は負論理で入力され、正論理に反転された AREA[3:0]信号と“CONFIG.TXT”上の 16 進数文字と比較が行われ、一致した 16 進文字と関連付けられたバイナリデータをコンフィギュレーションデータとして選択します。
- 表 4.2 はロータリースイッチ指定値と、スイッチに接続された XAREA[3:0]の信号レベルと、それによって指示されるエリア番号 の対応表です。

ロータリ SW 番号	XAREA				詳細
	3	2	1	0	
0	H	H	H	H	エリア 0
1	H	H	H	L	エリア 1
2	H	H	L	H	エリア 2
3	H	H	L	L	エリア 3
4	H	L	H	H	エリア 4
5	H	L	H	L	エリア 5
6	H	L	L	H	エリア 6
7	H	L	L	L	エリア 7
8	L	H	H	H	エリア 8
9	L	H	H	L	エリア 9
A	L	H	L	H	エリア A
B	L	H	L	L	エリア B
C	L	L	H	H	エリア C
D	L	L	H	L	エリア D
E	L	L	L	H	エリア E
F	L	L	L	L	エリア F

表 4.2 XAREA[3:0]とエリア番号表

4.6. XMODE ピン

- Ver1.905 より、SVF Interpreter ブロックが検出したエラーと、そのエラー発生行を、LED の点滅で表示できるようになりました。
- 詳細は、5.3. SVF エラー表示機能 で詳しく説明しています。

XMODEpin 論理(SW 設定値)	動作
0(ON)	エラー3bit + LINENo 24bit の表示
1(OFF)	エラー3bit の表示のみ

5. LED

- 59JTagStick には、モジュールの状態を表示するための LED(緑)が実装されています。

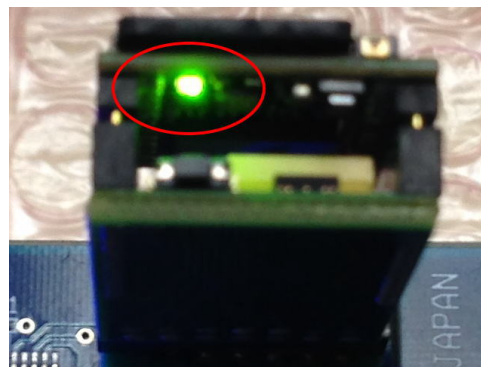



図 7.1 LED の実装位置

5.1. LEDの点灯条件

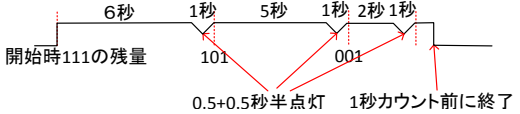
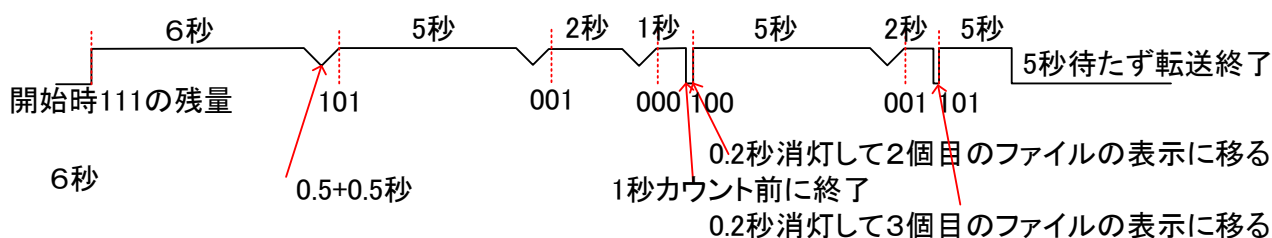
・ LEDの点灯条件は以下の通りです。

LED 状態	条件
点灯	microSD カードより JTAG ポートを介してデータを転送している間点灯します。 転送残量に応じて、周期的に LED が薄暗くなる区間を約1秒設け、点灯間隔により、大凡の残量を把握することができます。 残量の把握は、転送データファイルサイズの残量の上位3ビットをモニタして、以下のような間隔で暗くなります。 111 ： 6 秒全点灯＋1 秒半点灯 110 ： 6 秒全点灯＋1 秒半点灯 101 ： 5 秒全点灯＋1 秒半点灯 100 ： 5 秒全点灯＋1 秒半点灯 011 ： 4 秒全点灯＋1 秒半点灯 010 ： 3 秒全点灯＋1 秒半点灯 001 ： 2 秒全点灯＋1 秒半点灯 000 ： 1 秒全点灯＋1 秒半点灯 6秒 1秒 5秒 1秒 2秒 1秒 開始時111の残量 101 001 0.5+0.5秒半点灯 1秒カウント前に終了
点滅 167msec 点灯+ 167msec 消灯の繰り返し。	microSD カードのルートディレクトリに"CONFIG.TXT"を認識できなかった場合。 原因 ： "CONFIG.TXT"ファイルが存在しない。 スペルが間違っている。 "CONFIG.TXT"ファイルが 8 文字以上のキャラクタとして登録されている場合。(microSD カード上でコピペし、rename した場合など。)
	"CONFIG.TXT"によって指定されたバイナリデータがルートディレクトリ上で認識できなかった場合。 原因 ： バイナリデータが存在しない。 スペルが間違っている。 バイナリデータが 8 文字以上のキャラクタとして登録されている場合。(microSD カード上で 8 文字以下に rename した場合など。)
無点灯	microSD カードが挿入されていない場合は消灯のままです。
SVF(GVF) エラー表示	SVF(GVF)ファイルを指定した際に、SVF_Interpreter ブロックが検出したエラーと、そのライン行数を3ビットと 24 ビットの点滅によって表示します。(5.3 参照)
バージョン情報 表示	"CONFIG.TXT"で#P : C9 = 1 が指定されている場合。 上記の microSD カードが挿入されていない場合を除き、モードに関係なくバージョン情報が表示されます。

表 5.1 LED 点灯条件

5.2. 転送残量表示機能

- Ver1.905 より、転送データの残量を、LED の明滅の間隔により、把握することができるようになりました。
- 明滅は、0.5 秒かけて緩やかに暗くなり、その後、0.5 秒かけて緩やかに明るくなることにより、消灯と区別することができます。
- 転送データのファイル残量サイズの上位 3 ビットをモニタし、その値によって、以下の6段階の表示間隔で、残量の大凡の進捗状況を把握することができます。

111,110	6秒の全点灯＋1秒の半点灯
101,100	5秒の全点灯＋1秒の半点灯
011	4秒の全点灯＋1秒の半点灯
010	3秒の全点灯＋1秒の半点灯
001	2秒の全点灯＋1秒の半点灯
000	1秒の全点灯＋1秒の半点灯

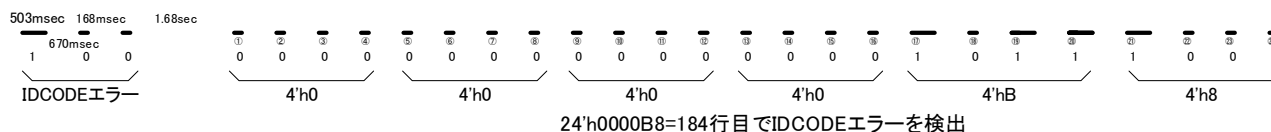
- 点灯開始は、6秒、若しくは、5秒の全点灯から始まり、その秒数以内に転送が終了した場合は、半点灯を行わず、転送終了時点で消灯します。
- ＋連結により、複数のファイルを転送する場合は、ファイルとファイルの切れ目に 0.2 秒の全消灯区間が入り、再び6秒、若しくは、5秒の全点灯から表示が始まります。
- 下記の図で、＋連結で3個のファイルを転送した場合の LED の表示例を示します。

5.3. SVF エラー表示機能

- Ver1.905 より、SVF_Interpreter ブロックで検出したエラー内容と、その行数を表示する機能を追加しました。
- エラーは3ビットの点滅の長さで表現し、その後に間隔を明けて、24ビットの点滅の長さで、約 1677 万行までのエラー箇所の表示が可能です。
- “CONFIG.TXT”内で、#P : CE = 0 と設定することにより、3ビットのエラー表示だけで、ライン行の表示がマスクされ、#P : CE = 1 と表示することにより、3ビットのエラー表示とそのライン行が表示されます。
- #P : CE = 0/1を指定しない場合、XMODE ピンにより、ライン行の表示の有無を切り替えることが可能です。

XMODEpin 論理(SW 設定値)	動作
0(ON)	エラー3bit + LINENO24bit の表示
1(OFF)	エラー3bit の表示のみ

- LINENO の表示の制御は、V1.904 以前で XMODE ピンが制御できない回路構成となっている場合はコマンドの制御により切り替えを行い、XMODE ピンを設定できる回路構成となっている場合は、XMODE ピンで切り替えが行えます。
- 表示途中で XMODE ピンを切り替えた場合、その切り替え変化を抽出し、そのモードに応じた表示を即時に再開します。
- エラー3ビットの表示内容は以下の通りです。

errbit	エラー内容	エラーの詳細
100	IDCODE エラー	IDCODE の比較で、期待した IDCODE と異なった IDCODE が読み出された。
010	INIT エラー	INIT の状態確認で、1 秒間モニタしても期待した値に変化しなかった。
011	DONE エラー	DONE の状態確認で、1 秒間モニタしても期待した値に変化しなかった。
001	TDO エラー	TDO の戻り値で、期待した値とは異なった値が返ってきた。
111	その他の SVF エラー	認識できないコマンドやパラメータを検出した。 何らかの理由で6秒以上、同じラインに停滞した。*1

- ERR3bit + LINENO24bit の表示例です。

5.4.　バージョン情報表示機能

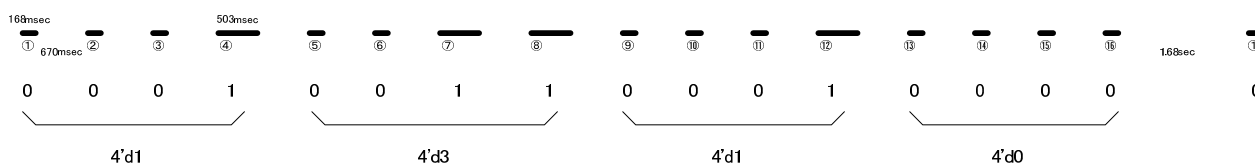
- V1.905 では、ゲート規模の制約のため、本機能を一時的に取り外しています。
- “CONFIG.TXT”で #P : C9 = 1 を指定することにより、本モジュールの裏面に実装された LED(図 5.1 参照)の点滅により、モジュールのバージョン情報を表示させることができます。
- バージョン情報は、16 ビットからなり、4 ビット毎に分割した 4 桁のヘキサデータとして表現されます。
- モールス信号のように、LED の点灯時間の長短で“0”と“1”の二進情報を表現します。
- 168msec の短い点灯と 670msec の消灯で“0”を表します。
- 503msec の 3 倍長い点灯と 670msec の消灯で“1”を表します。
- 16 回の点灯のあと、1.68sec の消灯区間で一巡したことを意味します。
- 一周目は無意味なデータが入っているため読み飛ばして、二周目から読みだして下さい。
- 図 5.2 は LED の点滅を時間系列上に太線で表した例です。時間は左から右に経過し、太線が点灯中を表し、太線と太線の間の空白は消灯区間を表します。
- LED の点滅の長さを見ながら“0”,“1”をメモしていくと、16 個の 2 進の繰り返しとなります。
  4 個ずつ区切ってバイナリをヘキサに変換すると、“1310”となります。
  これは現在の最新情報で、バージョンが“1h”、リビジョンが“31h”、サフィックスが“0h”であることを示します。
- バージョン情報は、大規模な変更が行われない限りこの数値が続きます。
- リビジョン情報は、論理の修正があるアップデードが行われるとインクリメントします。
- サフィックス情報は、同一論理において、インタフェース条件の異なるものが存在する場合に割り振られます。

Version　1. 310

図 5.2 LED 点滅例

6. “CONFIG.TXT”の準備

6.1. “CONFIG.TXT”とは

- microSD カードから FPGA をコンフィグレーションする時に参照されるテキストファイルです。
- ファイル名は 半角アスキーの“CONFIG.TXT”(小文字、大文字小文字混在も可)に固定化されており、microSD カードのルートディレクトリに置く必要があります。
- “CONFIG.TXT”ファイルには、ビットファイル名や、各種パラメータを列記します。
- “CONFIG.TXT”ファイルで取り扱えるキャラクタは、半角アスキーコードのみです。
- “CONFIG.TXT”ファイル内のコマンドは、大文字、小文字、両者混在で記載してもかまいません。
  例えば PO(ピーオー)など、Po と記載して 0(ゼロ)との紛らわしさを避けることができます。
  漢字やひらがなは使えません。全角の空白の混入には注意して下さい。

6.2.　Commands 速見表

第 1op	第 2op	Value	機能詳細
//			コメントアウト
#M:		A/X	メーカデフォルト指定
#S:		0〜D	TCK 速度選択
#P:	SS=	0/1	シリアル出力 MSB/LSB スワップ
	SB=	0/1	バイト出力 MSB/LSB スワップ
	SW=	0/1	Word,Dword 出力バイトスワップ
	PR=	xxxx_xxxx	プリアンブル長指定
	PO=	xxxx_xxxx	ポストアンブル長指定
	PM=	xxxx_xxxx	File+連結のミドルアンブル長指定
	D1=	xxxx_xxxx	nSTATUS〜Data 出力開始までのディレイ間隔
	D2=	xxxx_xxxx	XRST〜TCK 出力開始までのディレイ間隔
	PW=	xx	JTAG コマンド長指定
	C0=	0/1	ワードアライナ機能(ALTERA では使用しません）
	C1=	0/1	nCONFIG,~~nSTATUS~~　Pump ON
	C2=	0/1	P2to1 x 4port on SelectMap8
	C3=	0/1	P4to1 x 2port on SelectMap8
	C4=	0/1	DONE 信号無視
	C5=	0/1	FPP x4 モード ON
	C6=	0/1	XCS copy merge detect on
	C7=	0/1	外部回路 74138 の追加による XCS[3:0]＝＞XCS[7:0]拡張モード ON
	C8=	0/1	Future use
	C9=	0/1	Ver 表示(V1.905 では機能しません。)
	CA=	0/1	SVF 内の Frequency コマンドを無効にし、#S:コマンドの設定を使用する。
	CE=	0/1	SVF_Interpreter エラーのエラー行の表示の有無をコマンド指定する。
#R:		0〜F	リトライ回数
#0:		File1 + file2 +	エリア0ファイル指定
〜			エリア 1〜E ファイル指定
#F:		File1 + file2 +	エリア F ファイル指定

注)表中、グレイでハッチングのコマンドは、uSDCONF1A03JTag, 59JTagStick では無視されます。

6.3. Commands

(a) “/” (“slash”)

- コメントアウトを行います。
- 行中に “/” を検出すると、“CR”までのそれ以降の文字列をコメントとして読み飛ばします。
- サンプルでは、Verilog の慣例に従い、“//”で表記しています。

(b) #M : A/X

- Maker を指定します。デフォルトはありません。
- 本パラメータを指定することにより、7.4(2)のパラメータが自動で設定されます。
- デフォルトパラメータ値を変更したい場合、本コマンドの後に変更したいパラメータのコマンドを記述して下さい。設定値が上書き修正されます。
- PO パラメータなどの FPGA によって異なるパラメータだけ設定して下さい。

(c) #S : 0/1/2/3/4/5/6/78/9/A/B/C/D

- モジュールに書き込む JED ファイルレベルで、50MHz 版と 60MHz 版があります。
  ベータ版は 50MHz 版で開発を行い、安定したところで 60MHz 版もリリースします。
- FPGA 側の TCK の速度を指定します。
  - （50MHz 版）
    0 : 50MHz(デフォルト)
    1 : 25MHz
    2 : 12.5MHz
    3 : 6.25MHz
    4 : 3.57MHz
    5 : 2.778MHz
    6 : 1.923MHz
    7 : 1MHz

  - （60MHz 版）
    0 : 60MHz(デフォルト)
    1 : 30MHz
    2 : 15MHz
    3 : 7.5MHz
    4 : 5MHz
    5 : 3MHz
    6 : 2MHz
    7 : 1MHz
- SVF ファイル指定時、ファイル内の FREQENCY コマンドが指定する周波数がそのまま出力される訳ではなく、その値を超えない最も近い TCK 速度が選択されます。

(d) #P : SS = 0/1

- JTAG 版では JTAG の規格上、LSB ファーストに設定する必要があります。
- TDI ピンに最初に出力されるビットの MSB/LSB の出力順番の選択を行います。

  0 : MSB ファースト

  1 : LSB ファースト(デフォルト)

(e) #P : SB = 0/1

- 本コマンドは JTAG 版では未使用です。指定しても機能しません。

(f) #P : SW = 0/1

- JTAG 版では未使用です。指定しても機能しません。

(g) #P : PR = xxxx_xxxx

- バイナリデータを FPGA に送信する前に、プリアンブルデータとして Data Bus を All"1"にした状態で、指定回数の TCK を出力します。
- 指定は 16 進数で行い、8 桁全てを指定する必要があります。
  0000_0000 から FFFF_FFFF まで指定できます。
- 16 進数間にアンダーバー"_"を任意に入れることが可能です。
- 0000_0000 を指定すると、プリアンブルは出力されません。
- 指定がない場合のデフォルトは 0000_0000 です。(アルテラ FPGA はプリアンブルは不要なため、指定せず、デフォルトの 0000_0000 として下さい。)
- Altera の RBF ファイルの先頭にある 32 個の FF についてはデータと認識して出力されますので、本パラメータの指定の有無に係わらず出力されます。
- XILINX の bin ファイルの先頭にある 32 個の FF についてはデータと認識して出力されますので、本パラメータの指定の有無に係わらず出力されます。

(h) #P : PO = xxxx_xxxx

- バイナリデータを送信終了後、FPGA 内部の Initialaization が完了させるため、所定の TCK を送り続ける必要があります。
- バイナリデータを FPGA に送信した後に、ポストアンブルデータとして Data Bus を All"1"にした状態で、指定回数の TCK を出力します。
- 指定は 16 進数で行い、8 桁全てを指定する必要があります。
- 0000_0000 から FFFF_FFFF まで指定できます。
- 16 進数間にアンダーバー"_"を任意に入れることが可能です。
- 0000_0000 を指定すると、ポストアンブルは出力されません。
- FFFF_FFFF を指定すると、TCK は停止せずに出力されつづけます。
  この場合、 #R コマンドは機能しません。
- 何も指定しないと、デフォルトとして 0000_1000 が設定されます。
- DCLK は DONE 信号がアクティブになっても出力され続けます。

(i) #P : PM = xxxx_xxxx

- JTAG 版では未使用です。指定しても機能しません。

(k) #P : D1 = xxxx_xxxx

- JTAG 版では未使用です。指定しても機能しません。

(l)#P : D2 = xxxx_xxxx

- XRST が High になって TCK を出力するまでの間隔を指定します。
- 指定は 16 進数で行い、8 桁全てを指定する必要があります。
- 0000_0000 から FFFF_FFFF まで指定できます。
- 16 進数間にアンダーバー"_"を任意に入れることが可能です。
- 設定数値 x 20nsec がディレイ時間となります。(50MHz でカウントします。)
- SD カードからのコンフィギュレーションの場合、SD カード読み出し準備時間の 17msec より短い時間を設定した場合、機能しません。
- デフォルトは 000C_0000 が設定され、約 15.7msec の遅延が挿入され、SD カード読み出し準備時間による遅延による動作と、ほぼ等価となります。

(m)#P : PW = xx

- JTAG の IR レジスタの命令長を 16 進指定します。
- デフォルトは0A(10bit)です。
- JTAG 用の専用コマンドです。
- V1.8xx 以前のバージョンではコメントアウトする必要があります。
- デバイス毎の設定値は以下の通りです。
- Virtex7V2000T,7VX1140T,7VH580T,7VH870T につきましては、マルチコア品のため、現時点では対応できていません。(調査・デバッグ中です。)

	FPGA device	JTAG Instruction Length (bits)	Note
Altera	All devices	0A(10bit)	
Xilinx	*7 Family(Artix,Kintex,Virtex)*	06(6bit)	
	Virtex-7 (7V2000T, 7VX1140T, 7VH580T, 7VH870T)	調査・デバッグ中	
	Virtex-6	0A(10bit)	
	Virtex-5	0A(10bit)	
	Spartan-6	06(6bit)	
	Virtex-4	05(5bit)	
	Spartan-3	06(6bit)	

(n) #P : C0/1/2/3/4/5/6/7/8/9　= 0/1

・　0 から 9 のコントロールビットが定義されています。

① #P : C0 = 0/1　ワードアライナ機能

・　JTAG 版では未使用です。指定しても機能しません。

② #P : C1 = 0/1　　nCONFIG、Pump ON 指定

・　JTAG 版では未使用です。指定しても機能しません

・　JTAG 版では、他の機能に割り当てる可能性があります。

③ #P : C2 = 0/1

・　JTAG 版では未使用です。指定しても機能しません。

・　JTAG 版では、他の機能に割り当てる可能性があります。

④ #P : C3 = 0/1

・　JTAG 版では未使用です。指定しても機能しません。

・　TAG 版では、他の機能に割り当てる可能性があります。

⑤ #P : C4 = 0/1

・　JTAG 版では未使用です。指定しても機能しません。

・　TAG 版では、他の機能に割り当てる可能性があります。

⑥ #P : C5 = 0/1

・　JTAG 版では未使用です。指定しても機能しません。

⑦ #P : C6 = 0/1

・　JTAG 版では未使用です。指定しても機能しません。

⑧ #P : C7 = 0/1

・　JTAG 版では未使用です。指定しても機能しません。

⑨ #P : C8 = 0/1

・　未定義 (Reserved)

⑩ #P : C9 = 0/1

- V1.905 では機能しません。
- 本コマンドを ON にすると、動作モードに関わらず、モジュールのバージョン情報が LED の点滅により表示されます。
- 表示は 16 ビット単位で一巡し、長い点灯は“1”を、短い点灯は“0”を意味します。
- 先頭から 4bit 単位で区切って Hex 変換すると、モジュールのバージョン情報が得られます。
- 先頭の 4bit がバージョンを、次の 4bit とその次の 4bit の 2 桁でリビジョンを、最後の 4bit でサフィックスを表します。
- バージョン情報は、大きな変更があった場合に変更されます。
- リビジョンは、簡易な変更があった場合に変更されます。
- サフィックスは、同一バージョン、サフィックスにおいて、出力電圧などに違いを持たせた場合に割り付けられます。
- 表示方法の詳細は 8.3. バージョン情報表示機能の項目をご参照下さい。

⑪ #P : CA = 0/1

- 本コマンドを“1”にすると、SVF ファイル内の FREQUENCY コマンドの設定を無視し、#S: コマンドで指定された TCK 速度で動作します。
- デフォルトは”0”。

⑫ #P : CE = 0/1

- SVF インタプリターエラー発生時、エラー行の表示を行うか否かを制御します。
  - #P : CE = 0　　ERR3bit + ERRLINENO24bit の表示
  - #P : CE = 1　　ERR3bit のみの表示
- 本コマンドを指定しない場合、XMODE スイッチの論理によってエラー行の表示の有無を制御します。
  - XMODE スイッチ=”ON”　　ERR3bit + ERRLINENO24bit の表示
  - XMODE スイッチ=”OFF”　　ERR3bit のみの表示
- 本コマンドが設定された場合、コマンドが優先され、XMODE スイッチは機能しません。
- 本コマンドは、uSDCONF1A03JTAG モジュールを単独で使用し、XMODE ピンの論理が固定されていて、設定変更できない場合のために準備しています。
  59JTagStick では、本コマンドは指定せず、XMODE スイッチにより切り替えて下さい。
- XMODE ピンはエラー発生時に、運用状態で切り替えることで、エラー表示モードを瞬時に切り替えられます。

(o) #R : 0〜F

- JTAG では未使用です。指定しても機能しません。

(p) bitfile-name

- 上記の(a)または(b)に属さないキャラクタで始まる行は、Line-Processor はバイナリデータ名として扱います。(注：Line-Processor はキャラクタを解析するシーケンサ論理)
- バイナリデータは、“.rbf”フォーマットを指定して下さい。
- バイナリデータ名は 8 文字以下である必要があります。(拡張 FAT16 未対応)
- バイナリデータ名の前後にスペースや TAB を含むことができます。(ファイル名の間にスペースや TAB を挿入することはできません。)
- バイナリデータ名を”+”で連結することにより 8 個の FPGA まで CS 制御による Multi-FPGA コンフィギュレーションを行うことが可能です。 bitfile1.rbf + bitfile2.rbf + bitfile3.rbf + bitfile4.rbf
- “+”による連結の際、バイナリデータ名と”+”の間には必ず１つ以上の空白を挿入して下さい。
- ~~“+”による連結の際、途中に改行を加えないで下さい。~~
- V1.902 より、in-line にコマンドを記入することができるようになりました。詳しくは次章の in-line コマンドをご参照下さい。

(q)#0～#F : bitfile name

- バイナリデータ名を 0 から F の 16 個の 16 進数と関連付けを行います。
- 本コマンドにより関連付けが行われた場合、AREA ロータリーSW の指定エリアと同一の番号の関連付けが存在した場合、関連付けされたバイナリデータをコンフィギュレーションの対象とします。
- # の後には、0～9(30h～39h) , A～F(41h～46h) 若しくは a～f(61h～66h)を置くことができます。
- “:” の後に関連付けをしたいビットファイル名を記述します。
- バイナリデータは、ALTERA の場合、“.rbf”ファイルを指定して下さい。
- バイナリデータは、XILINX の場合、”bin”ファイル(“bit”も可能)を指定して下さい。
- バイナリデータ名は 8 文字以下である必要があります。(拡張 FAT16 未対応)
- バイナリデータ名の前後にスペースや TAB を含むことができます。(ファイル名の間にスペースや TAB を挿入することはできません。)
- バイナリデータ名を”+”で連結することにより最大8個までの Multi-FPGA コンフィギュレーションを行うことが可能です。

  #0:bitfile1.rbf + bitfile2.rbf + binfile3.rbf + binfile4.rbf
- “+”による連結の際、バイナリデータ名と”+”の間には必ず１つ以上の空白を挿入して下さい。
- V1.902 より、in-line にコマンドを記入することができるようになりました。詳しくは次章の”DaisyChain と in-line コマンドについて”をご参照下さい。
- V1.904 より、SVF ファイルをモジュールの書式に変換した GVF ファイルが扱えるようになりました。

6.4. DaisyChain と in-line コマンドについて

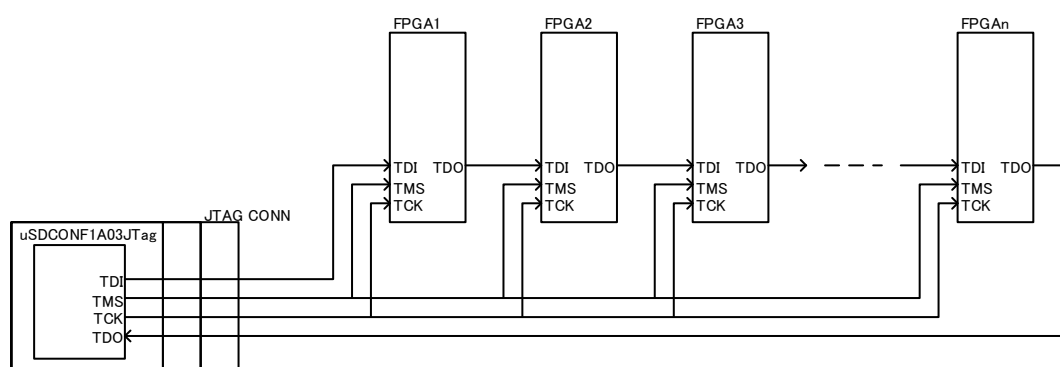
- V1.902 より in-line コマンドの記入ができるようになりました。これにより、異なるシリーズの FPGA のチェインや、メーカーの異なる FPGA の混在も可能になります。

6.4.1. DaisyChain の表記方法

- DaisyChain の記述方法は、ALTERA の FPP や PS、XILINX の SelectMap や SlaveSerial のときの記述と同一です。

  #0: fpga1.rbf + fpga2.rbf + fpga3.rbf + … + fpgan.rbf　　//ALTERA

  #1: fpga1.bin + fpga2.bin + fpga3.bin + … + fpgan.bin　　//XILINX

- 表記上の連結は、最大８個までです。
  後述する inDirect モード用のバイナリデータを列記する場合、合計で８個のファイル連結までで、FPGA の連結個数ではなくなります。
- 本機能は、組み込みが完了していますが、実機の都合で動作確認されていません。

6.4.2. In-line コマンドの表記方法

- 異なるシリーズの FPGA や、異なるメーカの FPGA を接続する場合、それぞれの FPGA に応じた設定を、転送前に行う必要があります。
- 以前のバージョンでは、この設定ができませんでしたが、V1.902 より可能になりました。
- この機能は、JTAG インタフェースに限らず、FPP,PS,SelectMAP,SlaveSerial の従来の FPGA インタフェースのモジュールについても同様に有効となります。
- ＋連結の表記のなかで、バイナリファイル名の前に括弧（）でコマンドを指定します。
- “)”と次のファイル名の間には、必ず空白を入れて下さい。
- inLine で指定したコマンドは、”)”に続くバイナリファイルに対してのみ有効で、そのバイナリファイルの転送が完了すると、デフォルト値や、リスト上部で設定された設定値に戻ります。
- コマンドは複数記述でき、各コマンドの最後にセミコロン”;”を追加すると改行として扱われ、同一行に羅列できます。
- 記述例(セミコロンを使った場合)

  #0 : fpga1.bin + (#M:A;) fpga2.rbf + (#M:X;) fpga3.bin

  #1 : fpga1.bin + (#p:pw=06;) fpga2.bin + (#p:pw=10;) fpga3.bin

- 記述例（セミコロンを使わない場合）

  #0 : fpga1.bin + (#M:A

  #p:d2=0000_0020

) fpga2.rbf + (#M:X

) fpga3.bin

6.4.3. FPGA をバイパスする方法

- FPGA をバイパスさせたい場合、予約語の”bypass”で始まる 8 文字以内のテキストファイル、”bypass X.txt”をバイナリデータの代わりに指定することでバイパスできます。
- ファイル名は大文字、小文字、その混在も可能です。
- 指定したファイル名のファイルは microSDCard に存在する必要がありますが、0 バイト以上のファイルであればよく、例えば、

  //fpga2 bypass

  などのように記入しておくと判別しやすくなります。
- 記入例

  #0：fpga1.bin + bypass2.txt + fpga3.bin + … + fpgan.bin
- 本機能は、実機の都合で動作確認されていません。

6.4.4. In-Direct モードの指定方法

- V1.905 ではサポートされていません。
- SPI や BPI に対して書込みを行うバイナリデータを予約語”bridge”で始まる 8 文字以下のバイナリファイル”bridgeX.bin/rbf”を用いることにより、SPI や BPI に対して ROM の中身の書き換えを行うことが可能です。
- 記述方法

  ＃0：bridge1.bin + fpga1.bin + bypass.txt + bypass.txt + … + bypass.txt

- 上記記述例では、1 個目の”bridge1.bin”で 1 個目の FPGA を立ち上げ、2 個目の”fpga1.bin”ファイルを、1 個目の FPGA 内の bridge1 論理と通信を行うことにより、SPI や BPI に”fpga1.bin”データを書込みます。
- SPI や BPI に書き込むためのデータは、bin,rbf や mcs,hex が指定できる予定です。
- 本機能はデバッグ中で、まだリリースされていません。
- bridgeX.bin/rbf ファイルは、機種依存となるため、別途費用が発生します。

6.5. SVF ファイルの対応

6.5.1. SVF ファイルの対応

- 59JTagStick(uSDCONF1A03JTag)では、Ver1.904 より SVF ファイルを指定することが可能になりました。
- SVF ファイルは、Serial Vector Format の略で、JTAG の TCK,TMS,TDI,TDO の４本の信号を使って、FPGA に固有のレジスタのアクセス方法や、コンフィグデータを含んだテキスト形式のファイルです。
- Programmer や iMPACT は、直接 JTAG ケーブルを経由して FPGA をコンフィグする代わりに、SVF ファイルに、JTAG の４本の信号のアクセス情報をベクタ形式で記憶させることができます。
- 本モジュールは、この SVF ファイルを翻訳でき、microSDCard に記憶された SVF ファイルを翻訳しながら JTAG ポート経由で FPGA への複雑な制御をエミュレートすることが可能です。
- ROM のイレースや、書込み、ベリファイなどにご利用いただけます。

6.5.2. SVF ファイルの対応コマンド

-

No.	コマンド	対応パラメータ	備考
１．	ENDDR	フルパラメータ	
２．	ENDIR	フルパラメータ	
３．	FREQUENCY	フルパラメータ	
４．	HDR	フルパラメータ	
５．	HIR	フルパラメータ	
６．	PIO	未対応	
７．	PIOMAP	未対応	
８．	RUNTEST	run_count TCK => FFFFFFFF まで SCK は未対応（TCK として処理されます。） min_time SEC => 85E00 SE（85 秒まで） 86E00 SEC でカウンタが一巡し、1E00 SEC に戻ることに注意	
９．	SDR	<length> TDI (tdi) TDO (tdo) MASK (mask) SMASK (smask) Length = 256 まで　256 を超えるコマンドは reformSVF で分割	
１０．	SIR	<length> TDI (tdi) TDO (tdo) MASK (mask) SMASK (smask) Length = 256 まで　256 を超えるコマンドは reformSVF で分割	
１１．	TDR	フルパラメータ	
１２．	TIR	フルパラメータ	
１３．	TRST	未対応	

6.5.3. GVF ファイルへの変換の必要性

- SVF ファイル内で、バイナリデータ情報は、SDR コマンドに記述されています。
- この SDR コマンドでは、巨大なバイナリデータの場合でも、長大な１行の HEX テキスト行で記述されており、長大なシフトレジスタにシフトインして、最後にラッチを行います。
- そこで扱われるデータの出現順序は、TDI に First　IN するデータがテキスト行の最後に出現する、いわゆる、Last IN/First OUT の順序です。
- このため、直接 SVF ファイルを取り込むためには、膨大なメモリが必要になるため、後述の reformSVF プログラムで、SDR コマンドを細かく分割し、並べ替えを行い、最小限のメモリで翻訳ができるよう、再編成し、GVF というサフィックスのファイルを生成します。

```
変換前
SDR 111836736 TDI (00000000000……ffffffff00000000) SMASK (ffffffffffffff……fffffffff) ;
                                                ⇐TDI
変換後
SDRS 256 TDI (0000ea100000000000000000000000000000000000000000ffffffff00000000) ;
                                                                     ⇐TDI
…….
SDRS 256 TDI (0000000000000000000000000000000000000000000000000000000000000000) ;
SDR 112 TDI (0000000000000000000000000000) ;
```

SDR 変換例

6.5.4. SVF,GVF ファイルの記述例

- 本モジュールは、SVF,GVF のどちらも読み込むことが可能です。
- 未変換の SVF ファイルについては、SDR のデータ長が 256 バイトを超えているとき、エラーを出力し、停止します。
- SVF,GVF ファイルは、rbf や bin/bit ファイルと同様の記述方式で、#0 から#F のエリア番号と転送データを関連付けします。
- ＋連結表記が可能で、連結された SVF(GVF)ファイルを順番に翻訳実行します。
- ＋連結により、変更したい SVF(GVF)ファイルだけ生成して、入れ替えることが可能です。
- 記述例

  #0：file1.svf

  #1：erase.gvf + prg_only.gvf　//イレーズ後プログラム

6.6. reformSVF.exe **の使い方**

- reformSVF.exe の使い方については、

  reformSVF 取扱説明書（reformSVF_how2use_rev1.pdf）

  にて、以下の３通りの方法について詳しく説明してありますので、そちらをご参照下さい。

  ① "送る"で変換する方法

  ② "SVF"ファイルを reformSVF プログラムと関連付けして変換する方法

  ③ ショートカットアイコンに Drag して変換する方法

6.7. D1,D2,PO の設定について

6.7.1. D1 パラメータ

・

・ FPP インタフェースを有する uSDCONF1A03 モジュールでは、D2 パラメータは、ALTERA のパラメータの tCFG(nCONFIG low pluse width)に該当し、2usec 以上です。
uSDCONF1A03 モジュールでは SDCard から config.txt とバイナリファイルの先頭を読み出し終わるまでの間、約 17msec 間 Low にして、動作可能になるのを待ちます。

・ JTag インタフェースを有する uSDCONF1A03JTag においては、nCONFIG/PROGB や nSTATUS/INITB のモニタが存在しないため、D2 パラメータによって、RST 解除から TCK を出力し始めるまでの時間おを指定できるようになっています。

・ D1 パラメータは、ALTERA のパラメータの tST2CK(nSTATUS hgih to first rising edge of DCLK)に該当しますが、JTAG 版では機能しません。
    ✧ 推奨地として D1=0000_0200(=10uSec)。config.txt でなにも指定しない場合、D1=0000_2000(=164usec)になります。

・ PO パラメータは、ALTERA パラメータの USRCLK 数とみなしてそれより大きい数値を設定して下さい。

・ 表 6.1 は、代表的な FPGA のシリーズのこれらのパラメータを列記した表です。

・ FPGA のシリーズにより、これらの値は異なり、共通に使える値はありません。このため、これらのパラメータを理解し、条件に合致した値を選択する必要がります。

・ 一覧にない FPGA をコンフィギュレーションする場合は、これらの値を調査し、最適な値を設定して下さい。

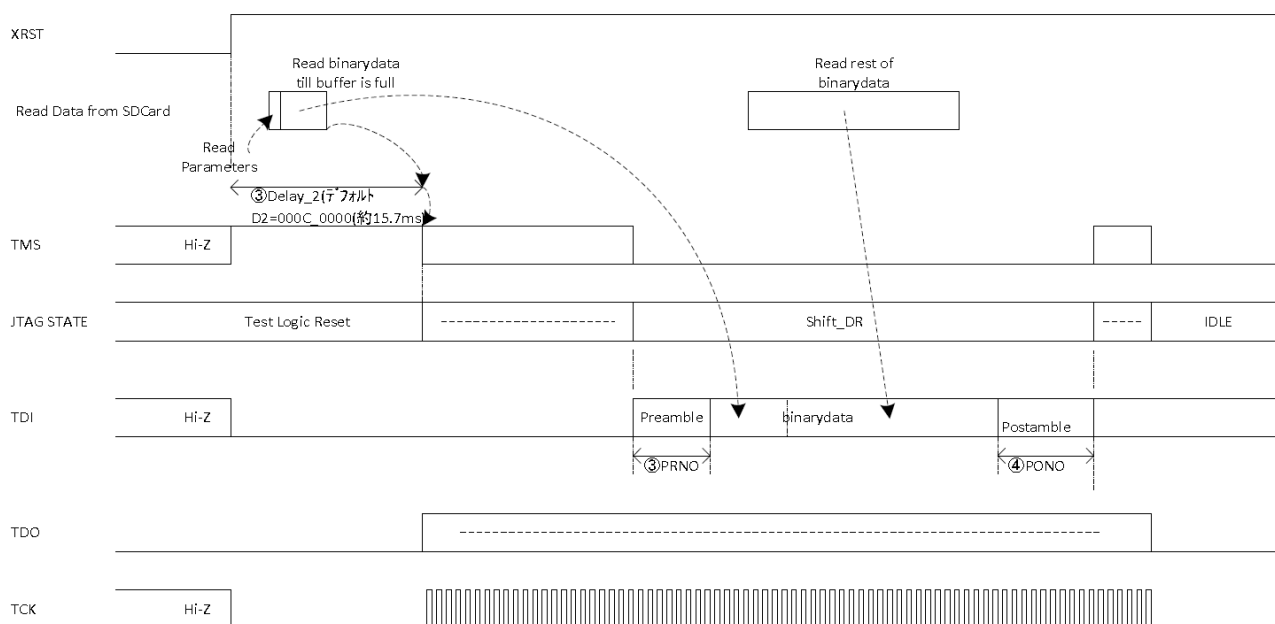



図 8.3 パラメータ位置関係

FPGA Series	tCF2ST1	tCF2CK	tST2CK	CLKUSR	D1	D2	PO
Cyclone-III	≦800uS	≧800uS	≧2uS	3,185/ 3,192(LS)	0000_0200 =10uS	000C_0000 =15.7mS	0000_1000 =4096clk
Cyclone-IV	≦230uS	≧230uS	≧2uS	3,192	0000_0200 =10uS	000C_0000 =15.7mS	0000_1000 =4096clk
Cyclone-V	≦1,506uS	≧1,506uS	≧2uS	17,408	0000_0200 =10uS	000C_0000 =15.7mS	0000_5000 =20480clk
Stratix-III	≦100uS	≧100uS	≧2uS	4,436	0000_0200 =10uS	000C_0000 =15.7mS	0000_1200 =4608clk
Stratix-IV	≦500uS	≧500uS	≧2uS	8,532	0000_0200 =10uS	000C_0000 =15.7mS	0000_2400 =9216clk
Stratix-V	≦1,506uS	≧1,506uS	≧2uS	17,408	0000_0200 =10uS	000C_0000 =15.7mS	0000_5000 =20480clk
Arria-GX	≦100uS	≧100uS	≧2uS	299	0000_0200 =10uS	000C_0000 =15.7mS	0000_0200 =512clk
Arria-II	≦500uS	≧500uS	≧2uS	8,532	0000_0200 =10uS	000C_0000 =15.7mS	0000_2400 =9216clk
Arria-V	≦1,506uS	≧1,506uS	≧2uS	17,408	0000_0200 =10uS	000C_0000 =15.7mS	0000_5000 =20480clk

表 6.5.1 ALTRA D1,D2,P0 設定例

FPGA Series	tPOR	tPL	tICCK	CLKUSR	D1	D2	PO
Virtex-5	10≦, ≦50ms	≦3ms	(400ns≦)		0000_0010 =320nS(def)	000C_0000 =15.7mS(def)	0010_0000 (def)
Virtex-6	15≦, ≦55ms	≦5ms	(400ns≦)		0000_0010 =320nS(def)	000C_0000 =15.7mS(def)	0010_0000 (def)
Virtex-7	10≦, ≦50ms	≦5ms	(150ns≦)		0000_0010 =320nS(def)	000C_0000 =15.7mS(def)	0010_0000 (def)
Kintex-7	10≦, ≦50ms	≦5ms	(150ns≦)		0000_0010 =320nS(def)	000C_0000 =15.7mS(def)	0010_0000 (def)
Artix-7	10≦, ≦50ms	≦5ms	(150ns≦)		0000_0010 =320nS(def)	000C_0000 =15.7mS(def)	0010_0000 (def)
Spartan-6	5≦, ≦40ms	≦4ms	(--)		0000_0010 =320nS(def)	000C_0000 =15.7mS(def)	0010_0000 (def)

表 6.5.2 XILINX D1,D2,P0 設定例

6.8. コマンドパラメータ　デフォルト値

・ “CONFIG.TXT”設定されるパラメータのデフォルト値を示します。

(1) メーカ指定無しの場合のデフォルト値

Maker Code	Maker Name	デフォルト設定値	備考
指定無		#P : SS = 0　(MSB ファースト) #P : SB = 1　(Swap 無し) #P : SW = 0　(Swap 無し) #S : 0　(50MHz Passive モード) #P : PR = 0000_0100　(プリアンブルなし) #P : PO = 0001_0000　(ポストアンブル 65536CLK 挿入) #P : PM = 0000_1000　(ミドルアンブル 4096CLK 挿入) ~~#P : D0 = 0000_1000　(nCONFIG ～ nSTATUS : 82usec)~~ #P : D1 = 0000_0010　(nSTATUS ～ DCLK : 320nsec) #P : D2 = 000C_0000　(XRST～nCONFIG : 15.7msec) #P : C0 = 1　(ワードアライナ有効) #P : C1 = 0　(Pump ON 無効) #P : C2 ,C3= 0　(Bus 分割 Multi FPGA mode 無効) #P : C9 = 0　(バージョン表示モード 無効)	

(2) メーカ指定をした場合のデフォルト値(ALTERA)

Maker Code	Maker Name	デフォルト設定値	備考
A	ALTERA	#P : SS = 1 (MSB ファースト) #P : SB = 0 (Swap 無し) #P : SW = 0 (Swap 無し) #S : 0 (50MHz Passive モード) #P : PR = 0000_0000 (プリアンブルなし) #P : PO = 0000_4000 (ポストアンブル 4096CLK 挿入) ~~#P : D0 = 0000_1000 (nCONFIG ～ nSTATUS : 82usec)~~ #P : D1 = 0000_2000 (nSTATUS ～ DCLK : 164usec) #P : D2 = 000C_0000 (XRST～nCONFIG : 15.7msec) #P : C0 = 0 (ワードアライナ無効) #P : C1 = 0 (Pump ON 無効) #P : C2 ,C3= 0 (Bus 分割 Multi FPGA mode 無効) #P : C9 = 0 (バージョン表示モード 無効)	

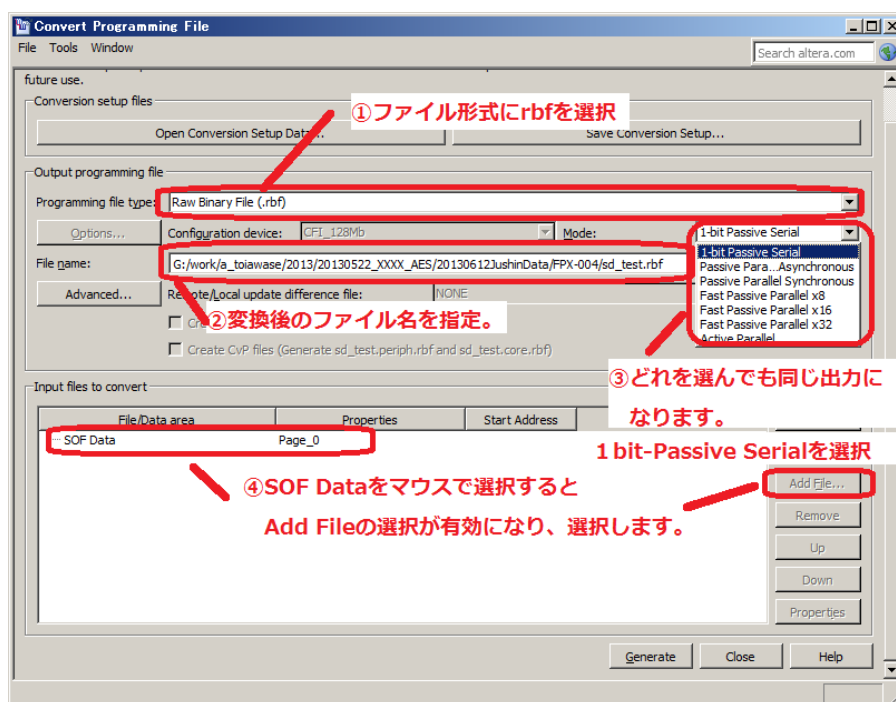
(3) メーカ指定をした場合のデフォルト値(XILINX)

Maker Code	Maker Name	デフォルト設定値	備考
X	Xilinx	#P : SS = 1 (LSB ファースト) #P : SB = 0 (Swap 無し) #P : SW = 0 (Swap 無し) #S : 0 (50MHz Slave モード) #P : PR = 0000_0100 (プリアンブル 256CLK 挿入) #P : PO = 0010_0000 (ポストアンブル 1048576CLK 挿入) ~~#P : D0 = 0000_1000 (PROGB ～ INITB : 82usec)~~ #P : D1 = 0000_0010 (INITB ～ CCLK : 320nsec) #P : D2 = 000C_0000 (XRST～PROGB : 15.7msec) #P : C0 = 1 (ワードアライナ有効) #P : C1 = 0 (Pump ON 無効) #P : C2,C3 = 0 (Bus 分割 Multi FPGA mode 無効) #P : C9 = 0 (バージョン表示モード 無効)	

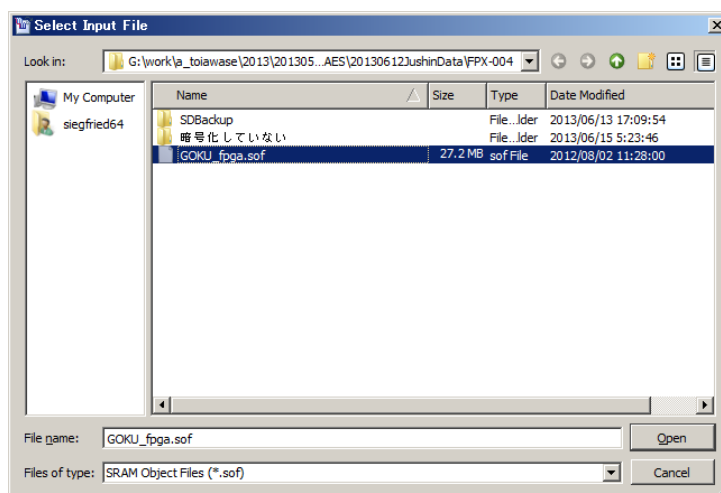
6.9. RBF データの生成方法

① Quartus-II･のの左上の PullDown メニューから、

File > Converter Program Files を選択します。

② 下記の Window が現れます。

③ ファイル形式に rbf を選択します。

④ 変換後のファイル名を指定します。8 文字以内で指定すると、そのまま SDCard にコピーが可能です。

⑤ モードを指定します。どれを選択しても rbf は同じ出力を吐き出しますので
とりあえず、1-bit Passive Serial を選択します。

⑥ ファイル名を選択し、Open をクリックします。

⑦ Generate をクリックすると rbf ファイルが出力されます。

⑧ 生成された rbf を SDCard のルートディレクトリにコピーします。

6.10. “CONFIG.TXT”サンプル

- microSD カードの root ディレクトリに、“CONFIG.TXT”(小文字も可)の名称のファイルを準備する必要があります。

6.10.1. アルテラの場合

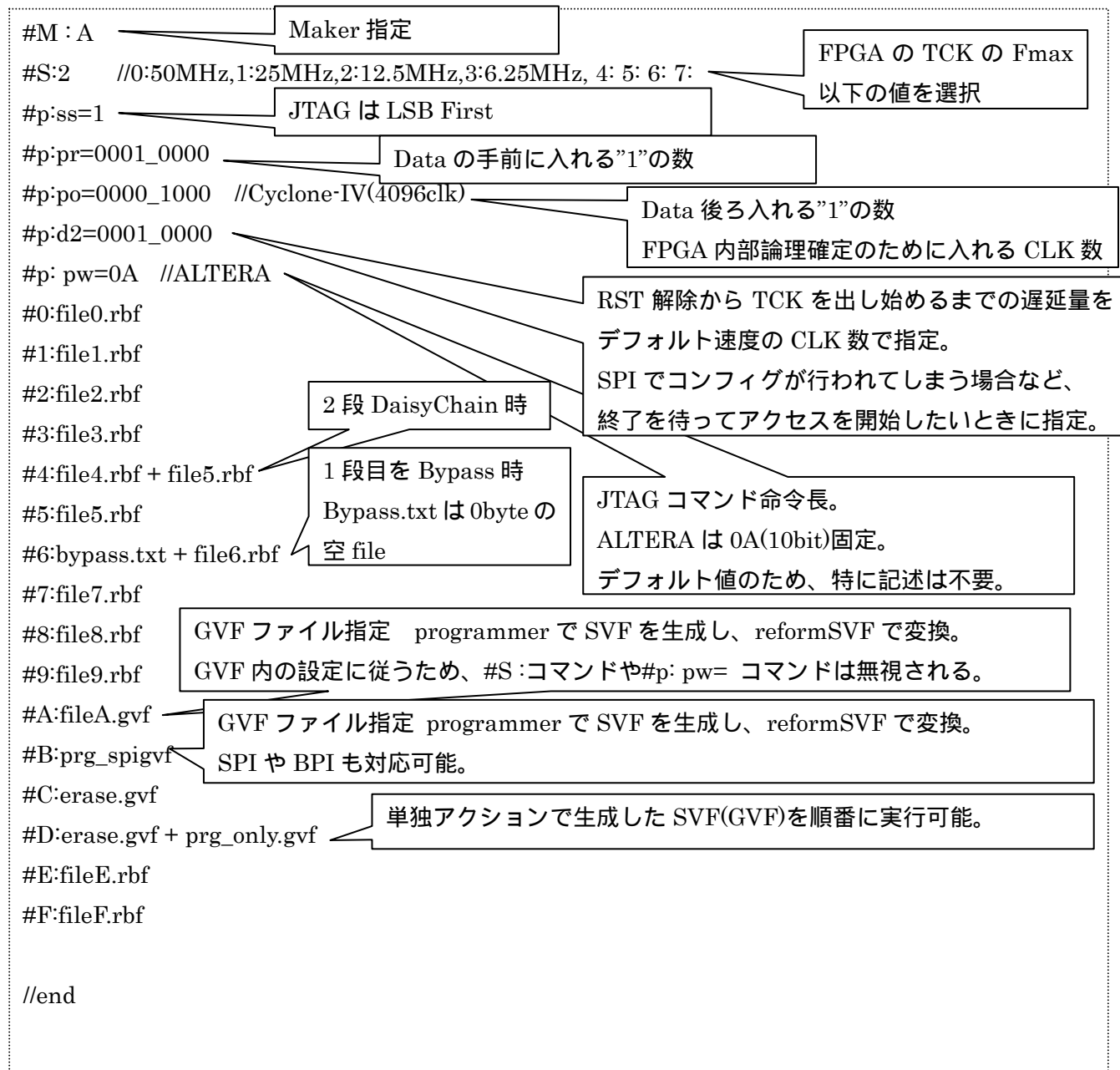
```
#M : A
#S:2      //0:50MHz,1:25MHz,2:12.5MHz,3:6.25MHz, 4: 5: 6: 7:
#p:ss=1
#p:pr=0001_0000
#p:po=0000_1000   //Cyclone-IV(4096clk)
#p:d2=0001_0000
#p: pw=0A   //ALTERA
#0:file0.rbf
#1:file1.rbf
#2:file2.rbf
#3:file3.rbf
#4:file4.rbf + file5.rbf
#5:file5.rbf
#6:bypass.txt + file6.rbf
#7:file7.rbf
#8:file8.rbf
#9:file9.rbf
#A:fileA.gvf
#B:prg_spigvf
#C:erase.gvf
#D:erase.gvf + prg_only.gvf
#E:fileE.rbf
#F:fileF.rbf

//end
```


図 6.10.1. “CONFIG.TXT”サンプル

6.10.2. XILINX の場合

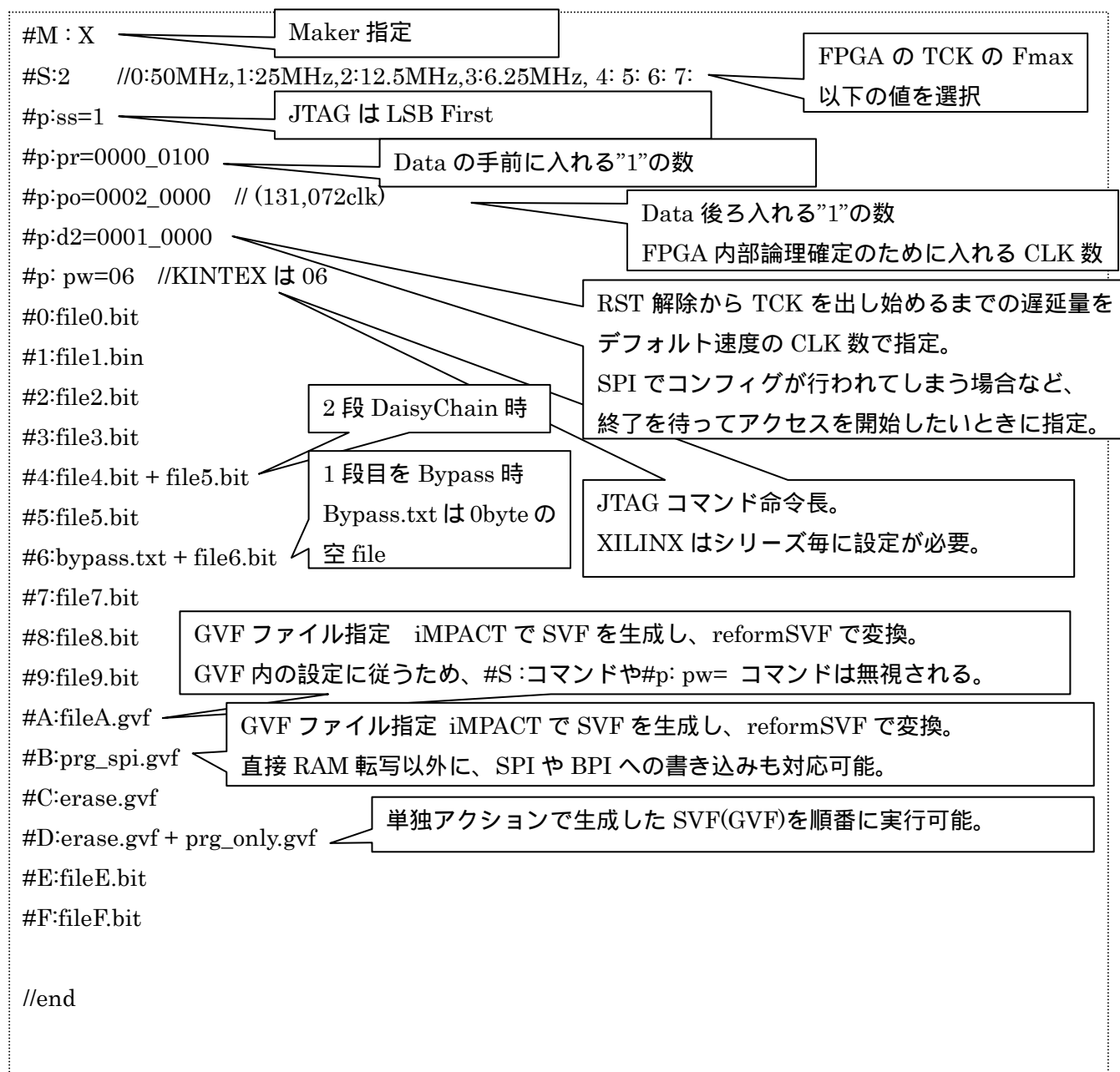



図 6.10.2. "CONFIG.TXT"サンプル

7. DCard の活線挿抜について

7.1. 構成図

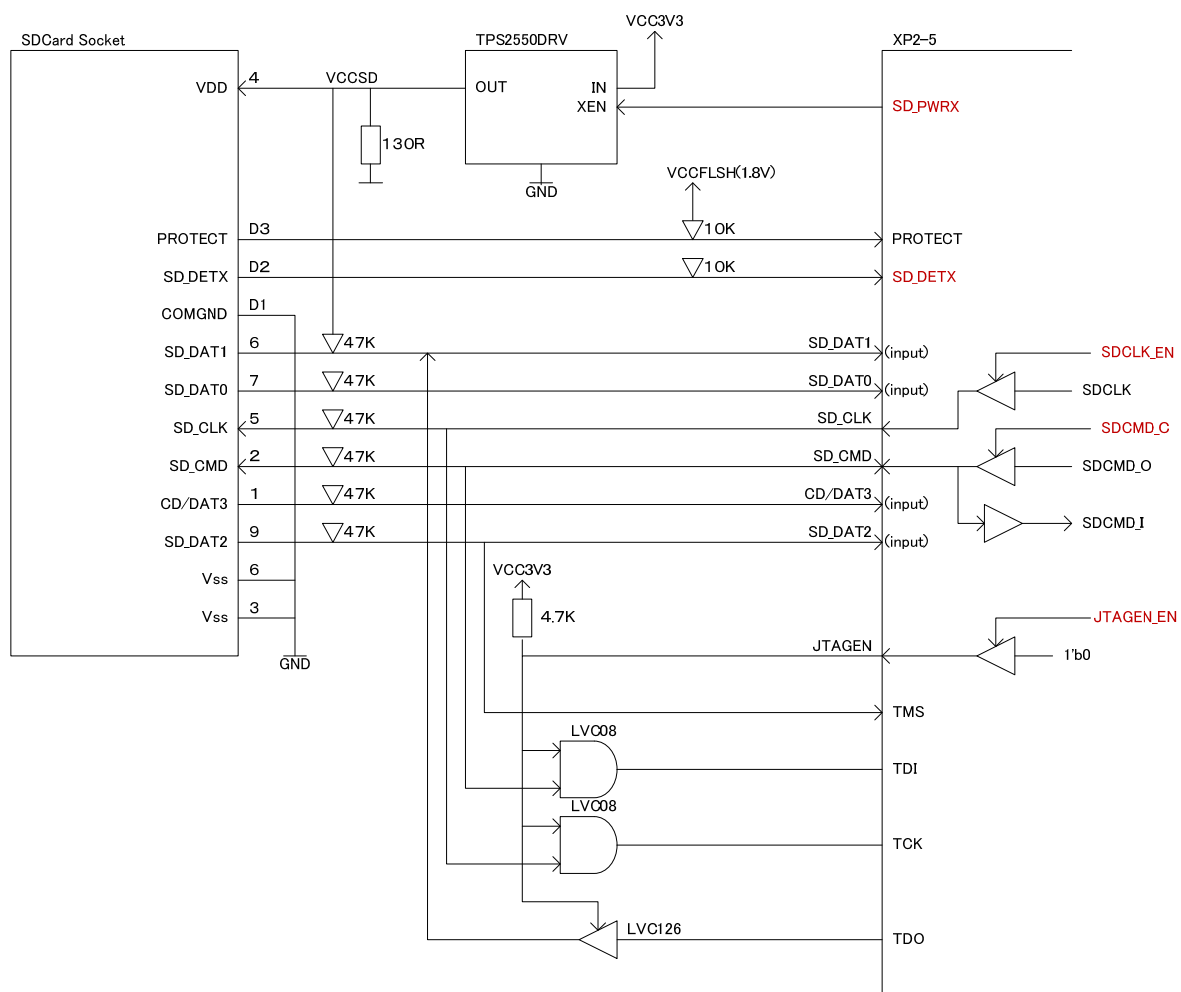



図 7.2.1 SDCard インタフェース部構成図

7.2. SDCard の活線挿抜対策

① SD カード電源の制御

- XRST 信号、SD_DETX 信号によって下記の制御を行ないます。

  assign SD_PWRX = XRST | SD_DETX;

- SD_DETX は、SDCONF モード、TRANS モード時の SDCard へのアクセスの最初の段階で1度だけチェックを行ないます。

  その際、SD_DETX == 1'b1で未挿入の場合、挿入を検出するまでループします。

- SD_DETX==1'b0を検出すると、7FFFh x 20nsec = 655usec 経過後、SDCard と通信を開始します。

  SDCard と通信中に SDCard が引き抜かれた場合、SDCONF3は ERROR ステートに遷移します。

  ERROR ステートに遷移した場合、電源を再投入するか XRST をアサートすることにより ERROR ステートから復旧することが可能です。

② IO ピンの Tri-State 制御

- SD_DAT[0:3]については、入力信号のため処理は行なっていません。
- SD_CMD については、通常入力方向の信号ですが、SDCard の挿抜時に出力に向いている場合を考慮して XRST,SD_DETX がアサート時、Hi-Z 出力となります。
- SD_CLK については、JTAG ポート選択時 Hi-Z 制御となる論理が入っておりますが、XRST,SD_DETX がアサート時も Hi-Z 出力となります。

③ JTAGEN 信号の制御

- 電源投入時、リセット立ち上がり時、SD_DAT0 が Low になっていないかをチェックします。

  もし SDCard_Adapter が挿入されていて、SD_DAT0 が Low に Pulldown されている場合、本モジュールは SDCard ソケットに JTAG ケーブルが挿入されたと判断し、JTAGEN 信号を Hi-Z とし、外部 4.7KΩPullup 抵抗により High に確定します。

- これにより、TDI,TCK が JTAG ポートに入力され、同時に TDO が SD_DAT1 ラインに出力されます。
- XP2 CPLD に論理が書かれていない状態では、JTAGEN 信号は Hi-Z と認識され、JTAG ポートの接続が確立します。
- XRST 動作時、SDCard が抜かれた状態において、SDCard への電源は停止状態になりますが、JTAG ポートへの SD_DAT1(TDO)信号が出力されないよう、JTAGEN は Low に固定されます。

8. モジュールのソケットからの取り外しについて

- uSDCONF1A03JTag はモジュール構成となっているため、ターゲット基板に実装する際にソケット実装することで、システムの評価が終了したあとに取り外して他のシステムに使いまわすことが可能です。
- しかしながら、図 8.1. のように、モジュールの裏側は高密度にチップ部品を実装しており、取扱いに注意して取り外さないと不用意に部品を押しつぶしたり、パターンを剥離してしまいかねません。
- 本章では、モジュールを末永くお使いいただくために、モジュールの裏面の実装の状況を把握していただき、取り外しの注意点等をご紹介いたします。

8.1. モジュール裏面

- "59kk"のシールの下に制御用のチップが実装されています。

  その両側の赤枠で囲った部分には、チップ抵抗やチップコンデンサ、ロジックチップが密集して実装されています。

図 8.1. uSDCONF1A03　Bottom View

8.2. モジュールの取り外し方

- ドライバ等でモジュールをソケットから抜き取る場合、図 8.1 の部品配置を常にイメージして、赤枠部分に力がかからないよう十分に注意して下さい。
- "59kk"のシールの貼られた制御チップの高さは 1.3mm あり、他の部品より背高です。

  ドライバ等で引き抜かれる場合は、こちらのチップに垂直に持ち上げる方向に力がかかるように、ゆっくりと、そして四隅が均等に持ち上がるように少しずつ操作を繰り返して下さい。
- 一気に持ち上げると、四隅のうちの一か所だけが持ち上がってしまってピンが曲がる原因になります。
- ドライバ等で取り外す場合、図 8.2 のようにモジュール下への差し込みが十分でない場合、図 8.1 の赤枠の部分に作用点が働いてしまうとチップ部品を破損してしまいます。

  また、下の基板に部品が配置されている場合、下の基板の部品を破損してしまう恐れがあります。
- 図 8.3 のように、ピンセットをモジュールの下を通し、両側から少しずつ持ち上げるとスムーズに取り外せます。

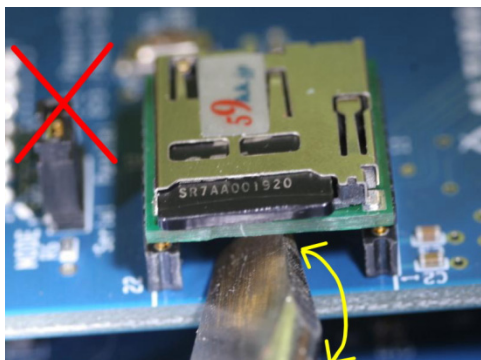



図 8.2 取り外しの悪い例

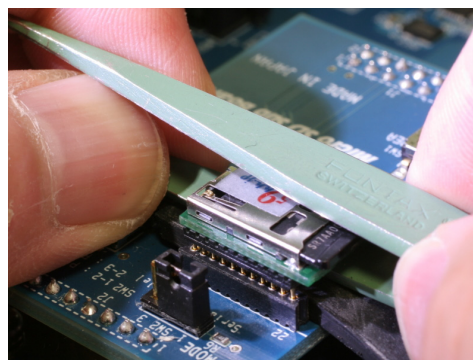



図 8.3 ピンセットを使った例

- 図 8.4 は D.I.Y で使用する“目地用コーキングヘラ”です。
- 図 8.5 はヘラをモジュール下に滑り込ませた写真です。
  ソケットの台座部分に滑り込んでいるのが判ります。
  この状態で両側から両手で少しずつ持ち上げると部品に接触することなく取り外しが可能です。
- こちらのヘラは、Amazon やコーナン PRO などでお買い求めいただけます。
  下記品名で検索をかけると Amazon やコーナン PRO が上位にヒットします。
  尚、モジュールご注文の際に弊社で調達することも可能です。ご入り用の際は弊社までお問い合わせください。
-

品名 ： 目地用コーキングヘラ曲り
発売元 ： 株式会社ハンディ･クラウン
商品コード ：309018 0000
注文型番 ：4905533-152782

図 8.4 目地用コーキングヘラ

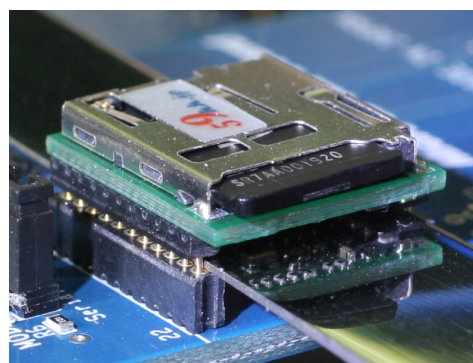



図 8.5 ヘラ使用例

9. 添付品

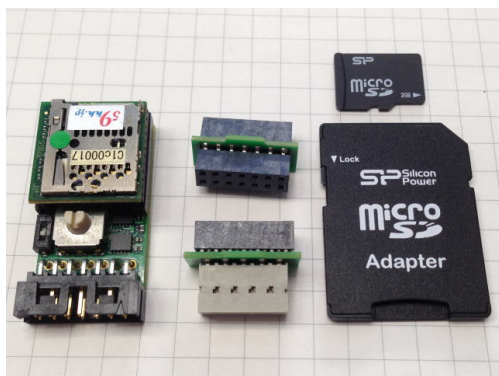



図 9.1

図 9.2

- 図 9.1 は製品として出荷される構成要素です。
  ① 59JTagStick 本体
  ② ALTERA 用変換コネクタ
  ③ XILINX 用変換コネクタ
  ④ 汎用先端分離タイプケーブルコネクタ
  ⑤ 2 Gbyte microSDCard（メーカはそのときの入手状況によって異なる場合があります。）
  SDCard の中の readme ディレクトリ内に、下記のデータが書き込まれています。
  (ア) Datasheet
  (イ) CONFIG.TXT サンプル(ALTERA、Xilinx 用)
  ⑥ SDCard アダプタ
  ⑦ 乾燥剤＋静電袋＋ブリスターケース
- 図 9.2 はケースに梱包された状態です。
  ケースのサイズは、突端部分を含めて 50mm × 85mm × 20mm　です。

10. ゲートサイズ、VCCIO 電圧とシールの色

- uSDCONF1A03JTag のゲートサイズ、VCCIO 電圧を識別できるよう、microSD ソケット表面に丸いシールが張られています。（表 13.1 参照）
- 59JTagStick では、ゲートサイズ８、VCCIO:2.5V 品、シール緑色が実装されています。

ゲートサイズ	VCCIO	ｲﾝﾀﾌｪｰｽ	シール
5	2.5V～3.3V	LVCMOS2.5V～3.3V	青色●
5	1.8V	LVCMOS1.8V	黄色●
8	2.5V～3.3V	LVCMOS2.5V～3.3V	緑色●
8	1.8V	LVCMOS1.8V	白色○

表 13.1

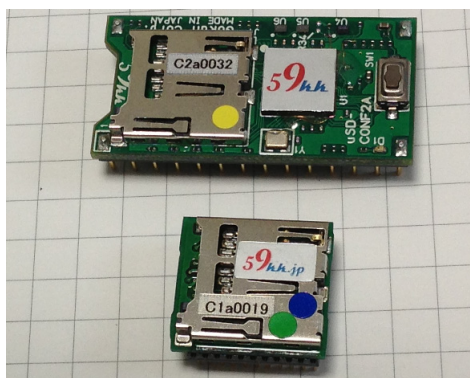



図 13.1　シール添付例

11. 制約事項

① SDカードは、添付の2Gバイト品をご使用下さい。

市販の2Gバイト以下の FAT16 でフォーマットされた SD カードもご利用いただけますが、初期のころの SD カードインタフェース速度が極端に遅いものや、最近のものでも、極まれに相性が合わない場合があります。

この場合、SDカードからのコンフィギュレーション時、転写時にLEDが点灯したままの状態になります。

FAT32でフォーマットされている2Gバイトを超えるSDHC品には対応していません。

2Gバイト以下のmicroSDカードでも、FAT32で再フォーマットされたものは動作しません。

再フォーマットを行う場合は FAT16(Microsoft Windows のボリュームのプロパティのファイルシステムでは FATと表示されます)を選択して下さい。

② ビットファイル名は英数字の8文字以下にして下さい。

拡張FAT16には対応していません。

大文字小文字どちらも使用できますが、識別は行いません。

"_"(アンダーバー)、"-" (ハイフン) も文字として使用できます。

尚、8文字以上のファイルを一旦microSDカードにコピーして、renameによりファイル名を8文字以下に修正させた場合、修正後も依然8文字以上のファイルとして取り扱われてしまい、先頭の7文字しかファイル名の識別要素として機能致しませんので、パソコンでファイル名を修正後、コピーを行って下さい。

③ "CONFIG.TXT"ファイルやバイナリデータのmicroSDカードへのコピーは、Windowsシステムより行って下さい。

Linux や Unix システム上で microSD カードにコピーすると、正しく動作しません。

(Linux や Unix 上で生成されたバイナリデータを、Windowsシステムを介してコピーする分には問題ありません。)

④ Virtex-7 のマルチコアタイプ、7V2000T,7VX1140T,7VH580T,7VH870T はデバッグ中で対応できていません。

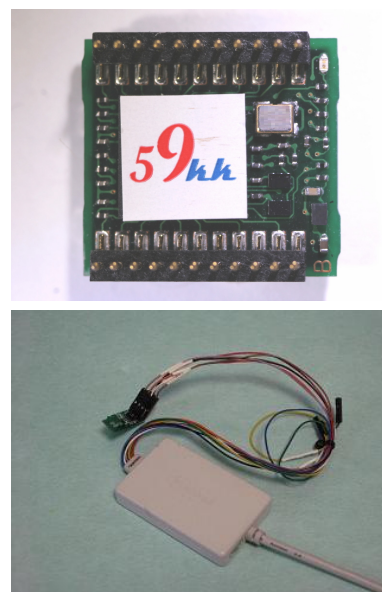
⑤ FPGAに接続されたBPIやSPIROMの書き換え機能は、近日対応予定です。

⑥ SVFインタプリタ機能も対応予定です。

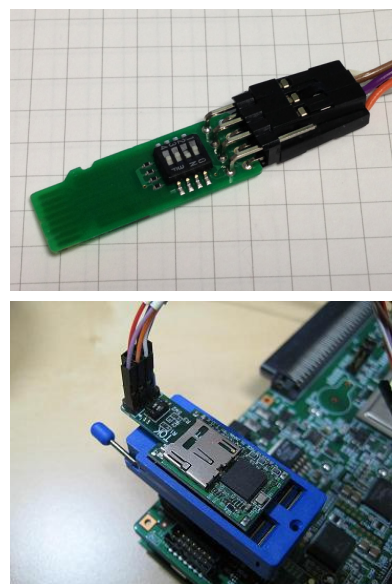
12. アップデート機能(順次対応予定)

“59kk”のシールの貼られたモジュールは、128 ビットの暗号 Key によって Encryption がかけられています。msd-Adapter と JTag ケーブルをご用意いただくことにより、同じ暗号 Key によって暗号化された bit ファイルを悟空株式会社のホームページよりダウンロードしてアップデートしていただくことが可能です。

“59kk”のシールの貼られていないモジュールにつきましては、暗号 Key の書き込みが必要ですので、弊社もしくは担当営業にご相談下さい。

(JTag ケーブルは、悟空株式会社にても販売する予定です。)

今後、下記のようなアップデートを予定しています。

- 機能改版 (一部有償)
  - FAT32 対応
  - SDHC 対応
  - コンフィギュレーション終了後の microSD カードへのインタフェース機能
  - SPI(Master/Slave)終端機能
  - JTag 終端機能
    - JTag からのコンフィギュレーション
    - ROM データの書き換え
    - 別の uSDCONF の書き換え

  など

以上

お問い合わせ等連絡先
悟空株式会社
担当 ： 大庭 （ オオバ ）
〒 220-0004
横浜市西区北幸 1-11-1 水信ビル7階
FreeCall : 0800-7775559 (平日 9:00～18:00)
電話 ：045-590-6227 Fax : 050-3156-1404
Email : info01@59kk.jp URL : https://www.59kk.jp